ML Lab Assessment - 3

Property Hunting - Malibu Beach-houses

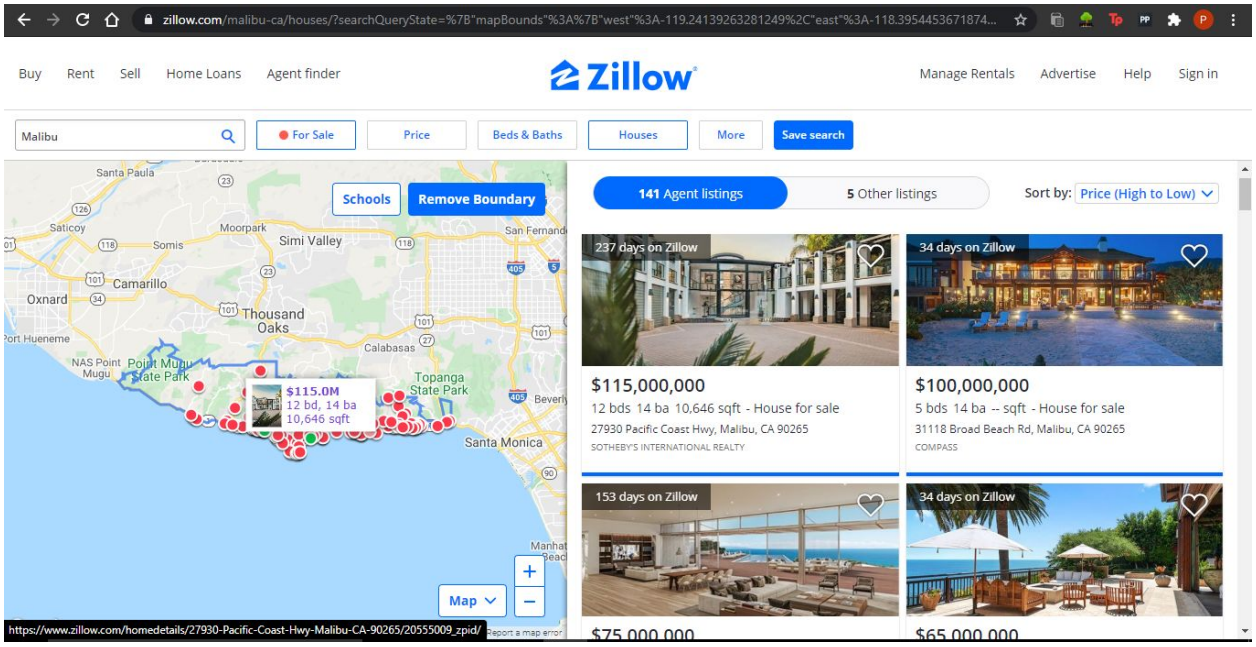
Team 1

18BCB0001	Vaibhav Vijay
18BCB0002	Anish S Ghiya
18BCB0015	Aditi Ranganath
18BCB0024	Siddharth M
18BCB0030	Prateek Chaturvedi
18BCB0102	Tanya Warrier
18BCB0108	Saksham Dewan

Video link of team discussion
https://drive.google.com/file/d/1ZPQmYY_vUkEEznpTPU6KmZl27hKz5TR-/view?usp=sharing

Step 1 - Finding a Data Source

https://www.zillow.com/malibu-ca/



Step 2 - Selecting features

1. Location/Name of Property
2. Price in USD
3. Number of Bedrooms
4. Number of Bathrooms
5. Lot size in square feet
6. Year Built
7. Minimum Estimated sales range (Estimated by the website itself)
8. Maximum Estimated sales range (Estimated by the website itself)

Step 3 - Consolidating Data

https://docs.google.com/spreadsheets/d/1Pv_CDkWxMgFtMk1IzronzKdyCBJuwlsWg5V87IMlCH4/edit#gid=0

	A	B	C	D	E	F	G	H
1	Location	Price ($)	Bedrooms	Bathrooms	Lot_Size (sqft)	Year Built	min_price($)	max_price($)
2	27930 Pacific Coast Hwy, Malibu, CA 90265	115,000,000.00	12	14	111,784.00	1997	12,300,000.00	21,530,000.00
3	31118 Broad Beach Rd, Malibu, CA 90265	100,000,000.00	5	14	51,382.00	2020	10,650,000.00	21,470,000.00
4	24186 Case Ct, Malibu, CA 90265	75,000,000.00	5	7	113,678.00	2021		
5	22102 Pacific Coast Hwy, Malibu, CA 90265	22,950,000.00	4	4	11,137.00		8,590,000.00	15,700,000.00
6	33256 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	3	4	114,569.00		4,100,000.00	6,180,000.00
7	24834 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	5	8	53,312.00	2016	8,400,000.00	13,500,000.00
8	5046 Carbon Beach Ter, Malibu, CA 90265	49,995,000.00	5	7	747,291.00	2019	12,700,000.00	21,460,000.00
9	23800 Malibu Crest Dr, Malibu, CA 90265	49,500,000.00	7	10	148,290.00	2019	42,910,000.00	48,450,000.00
10	33740 Pacific Coast Hwy, Malibu, CA 90265	45,000,000.00	6	8	74,945.00	2013	50,900,000.00	8,100,000.00
11	31272 Broad Beach Rd, Malibu, CA 90265	42,000,000.00	6	9	47,374.00	2004	34,870,000.00	39,750,000.00
12	21528 Pacific Coast Hwy, Malibu, CA 90265	40,000,000.00	5	8	12,646.00	2008	6,020,000.00	9,650,000.00
13	3093 Sweetwater Mesa Road, Malibu, CA 90265	39,500,000.00	5	7	1,708,563.00	2019	11,260,000.00	20,030,000.00
14	3903 Carbon Canyon Rd, Malibu, CA 90265	35,000,000.00	6	9	141,737.00	2009	31,370,000.00	35,040,000.00
15	30385 Morning View Dr, Malibu, CA 90265	29,995,000.00	8	9	177,263.00	2020	14,010,000.00	33,790,000.00
16	32554 Pacific Coast Hwy, Malibu, CA 90265	29,500,000.00	5	6	82,393.00	2000	5,290,000.00	8,820,000.00
17	23334 Malibu Colony Rd, Malibu, CA 90265	29,000,000.00	5	8	13,091.00	1971	26,280,000.00	29,050,000.00
18	3903 Carbon Canyon Rd, Malibu, CA 90265	35,000,000.00	6	9	141,737.00	2009	31,370,000.00	35,040,000.00
19	31134 Broad Beach Rd, Malibu, CA 90265	32,000,000.00	4	6	46,243.00	2011	28,990,000.00	32,040,000.00
20	23754 Malibu Rd, Malibu, CA 90265	31,500,000.00	4	6	11,974.00	1988	69,600,000.00	12,220,000.00
21	3605 Noranda Ln, Malibu, CA 90265	16,500,000.00	6	6	218,633.00	2021	42,000,000.00	6,670,000.00
22	28926 Cliffside Dr, Malibu, CA 90265	27,500,000.00	4	5	45,543.00		10,500,000.00	27,500,000.00

Step 4 - Feature Engineering Tasks

(i) Removing Commas

Prices were in the form of Strings, and ‘,’ is a special character so it cannot be recast into float data type until commas were entirely taken off.

(ii) Changing the data types

Prices: obj to float
Year Built: float to int
Lot_size: obj to float

(iii) Replacing missing data

Year Built: Replacing missing year with “1000”
Prices: Replacing missing values with “0”

(iv) Imputation

Prices: Replacing value “0” with mean price of each column

(v) One hot encoding

Year Built: Encoding the values of Year Built with {0,1}

Step 5 - Plotting and Visualization of data

Using Seaborn and pyplot for Visualisation of various features

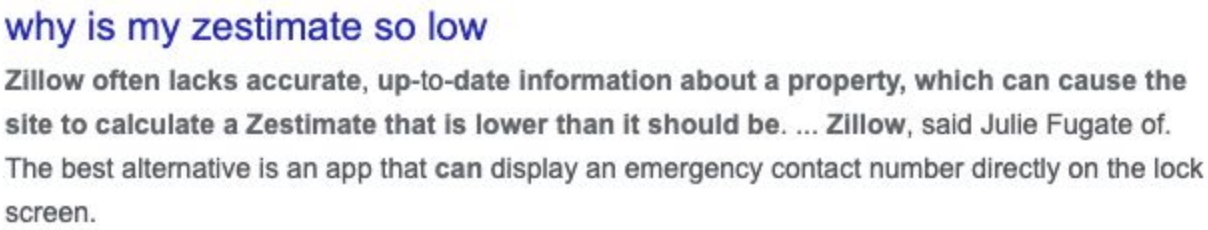
(i) Plot of Lot Size vs Price data and then a regression model fit on it. A secondary plot where the Plot size was limited to 60,000 sqft for better visualisation
(ii) Barplots for

- Bedroom vs Price
- Bedroom vs Bathroom

(iii) Boxplot for Bedroom vs Price to visualise the range of Price for a specific number of bedrooms
(iv) Year built, Lot size vs Price Data plot and a regression model fit on it.
(v) Checking Normality in Data
(vi) Correlation Heatmap

Step 6 - Bonus Task

As seen in the dataset, the actual ranges that the houses are priced on, are much higher than the maximum estimated price.
Well, even Google agrees



This is the problem that we could focus on, that can be subjected to a Machine Learning Analysis.
A machine learning model can be prepared to predict accurate prices based on all the key attributes that have been selected.
The dataset could definitely use more features to help with the prediction. Details on the sellers, their age, occupation and average income would help the model learn better.

Also, if possible , a ML analysis could be done to understand why people need more bathrooms than bedrooms

TEAM 1

Malibu Beachhouse Properties

The dataset

Importing the required dependancies

In [ ]:
```
import pandas             as pd
import numpy              as np
import matplotlib.pyplot as plt
import seaborn            as sns

from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
```

In [ ]:
```
malibu_dataset = pd.read_csv('/content/ZillowTabulate.csv')
```

Malibu Beachhouse Properties

In [ ]:
```
malibu_dataset.head()
```

Out[ ]:

	Location	Price ($)	Bedrooms	Bathrooms	Lot_Size (sqft)	Year Built	min_price($)	max_price(
0	27930 Pacific Coast Hwy, Malibu, CA 90265	115,000,000.00	12	14	111,784.00	1997.0	12,300,000.00	21,530,000.(
1	31118 Broad Beach Rd, Malibu, CA 90265	100,000,000.00	5	14	51,382.00	2020.0	10,650,000.00	21,470,000.(
2	24186 Case Ct, Malibu, CA 90265	75,000,000.00	5	7	113,678.00	2021.0	NaN	Na
3	22102 Pacific Coast Hwy, Malibu, CA 90265	22,950,000.00	4	4	11,137.00	NaN	8,590,000.00	15,700,000.(
4	33256 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	3	4	114,569.00	NaN	4,100,000.00	6,180,000.(

In [ ]:
```
malibu_dataset.dtypes
```

Out[ ]:
```
Location            object
Price ($)           object
Bedrooms             int64
Bathrooms            int64
Lot_Size (sqft)     object
Year Built         float64
min_price($)        object
max_price($)        object
dtype: object
```

A lot of discrepancies in datatypes were observed as we imported the data from the spreadsheet csv format to pandas dataframe which we will attempt to fix in the feature engineering secion of the code

In [ ]:
```
malibu_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Location         27 non-null     object 
 1   Price ($)        27 non-null     object 
 2   Bedrooms         27 non-null     int64  
 3   Bathrooms        27 non-null     int64  
 4   Lot_Size (sqft)  27 non-null     object 
 5   Year Built       24 non-null     float64
 6   min_price($)     26 non-null     object 
 7   max_price($)     26 non-null     object 
dtypes: float64(1), int64(2), object(5)
memory usage: 1.8+ KB
```

The 27 rows of the data occupy 1.8 kb of data

In [ ]:
```
malibu_dataset.describe()
```

Out[ ]:

	Bedrooms	Bathrooms	Year Built
count	27.000000	27.000000	24.000000
mean	5.444444	7.555556	2001.458333
std	1.694637	2.470337	20.308446
min	3.000000	4.000000	1949.000000
25%	5.000000	6.000000	1990.250000
50%	5.000000	7.000000	2009.000000
75%	6.000000	9.000000	2019.000000
max	12.000000	14.000000	2021.000000

Important to notice that the maximum of 14 bathrroms in a house and only 12 bedrooms as the maximum

Feature engineering

Variable datatype Corrections

Taking off the Commas from prices

In [ ]:
```
malibu_dataset['min_price($)']=malibu_dataset['min_price($)'].fillna("0"
)
malibu_dataset['max_price($)']=malibu_dataset['max_price($)'].fillna("0"
)
malibu_dataset['Year Built']=malibu_dataset['Year Built'].fillna(1000)
```

In [ ]:
```
x,y,z,t=[],[],[],[]
for i in range(malibu_dataset.shape[0]):
  x.append(malibu_dataset['min_price($)'][i].replace(',',''))
  y.append(malibu_dataset['max_price($)'][i].replace(',',''))
  z.append(malibu_dataset['Price ($)'][i].replace(',',''))
  t.append(malibu_dataset['Lot_Size (sqft)'][i].replace(',',''))
```

In [ ]:
```
malibu_dataset.head()
```

Out[ ]:

	Location	Price ($)	Bedrooms	Bathrooms	Lot_Size (sqft)	Year Built	min_price($)	max_price(
0	27930 Pacific Coast Hwy, Malibu, CA 90265	115,000,000.00	12	14	111,784.00	1997.0	12,300,000.00	21,530,000.(
1	31118 Broad Beach Rd, Malibu, CA 90265	100,000,000.00	5	14	51,382.00	2020.0	10,650,000.00	21,470,000.(
2	24186 Case Ct, Malibu, CA 90265	75,000,000.00	5	7	113,678.00	2021.0	0	
3	22102 Pacific Coast Hwy, Malibu, CA 90265	22,950,000.00	4	4	11,137.00	1000.0	8,590,000.00	15,700,000.(
4	33256 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	3	4	114,569.00	1000.0	4,100,000.00	6,180,000.(

In [ ]:
```
malibu_dataset['min_price($)'] = x
malibu_dataset['max_price($)'] = y
malibu_dataset['Lot_Size (sqft)'] = t
malibu_dataset['Price ($)'] = z
malibu_dataset['min_price($)'] = malibu_dataset['min_price($)'].astype(f
loat)
malibu_dataset['max_price($)'] = malibu_dataset['max_price($)'].astype(f
loat)
malibu_dataset['Price ($)'] = malibu_dataset['Price ($)'].astype(float)
malibu_dataset['Lot_Size (sqft)'] = malibu_dataset['Lot_Size (sqft)'].as
type(float)
```

In [ ]:
```
malibu_dataset['min_price($)']=malibu_dataset['min_price($)'].fillna("0"
)
malibu_dataset['max_price($)']=malibu_dataset['max_price($)'].fillna("0"
)
malibu_dataset['min_price($)']=malibu_dataset['min_price($)'].astype('fl
oat')
malibu_dataset['max_price($)']=malibu_dataset['max_price($)'].astype('fl
oat')
malibu_dataset['Year Built']=malibu_dataset['Year Built'].fillna(1000)
malibu_dataset['Year Built']=malibu_dataset['Year Built'].astype('int')
```

In [ ]:
```
imputer = SimpleImputer(missing_values = 0.0, strategy = 'mean')
imputer.fit(malibu_dataset.iloc[:,6:8])
malibu_dataset.iloc[:,6:8] = imputer.transform(malibu_dataset.iloc[:,6:8
])
pd.options.display.float_format = '{:,.2f}'.format
```

In [ ]:
```
scaler = MinMaxScaler()
malibu_dataset['Scaled Price']=scaler.fit_transform(np.array(malibu_data
set['Price ($)']).reshape(-1,1))
```

In [ ]:
```
malibu_dataset.head()
```

Out[ ]:

	Location	Price ($)	Bedrooms	Bathrooms	Lot_Size (sqft)	Year Built	min_price($)	max_price($)
0	27930 Pacific Coast Hwy, Malibu, CA 90265	115,000,000.00	12	14	111,784.00	1997	12,300,000.00	21,530,000.00
1	31118 Broad Beach Rd, Malibu, CA 90265	100,000,000.00	5	14	51,382.00	2020	10,650,000.00	21,470,000.00
2	24186 Case Ct, Malibu, CA 90265	75,000,000.00	5	7	113,678.00	2021	21,042,310.85	23,013,076.92
3	22102 Pacific Coast Hwy, Malibu, CA 90265	22,950,000.00	4	4	11,137.00	1000	8,590,000.00	15,700,000.00
4	33256 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	3	4	114,569.00	1000	4,100,000.00	6,180,000.00

Filling missing values

Replacing missing years

In [ ]:
```
malibu_dataset['Year Built']=malibu_dataset['Year Built'].astype(int)
```

Imputation to replace zero values of prices

In [ ]:
```
malibu_dataset.head()
```

Out[ ]:

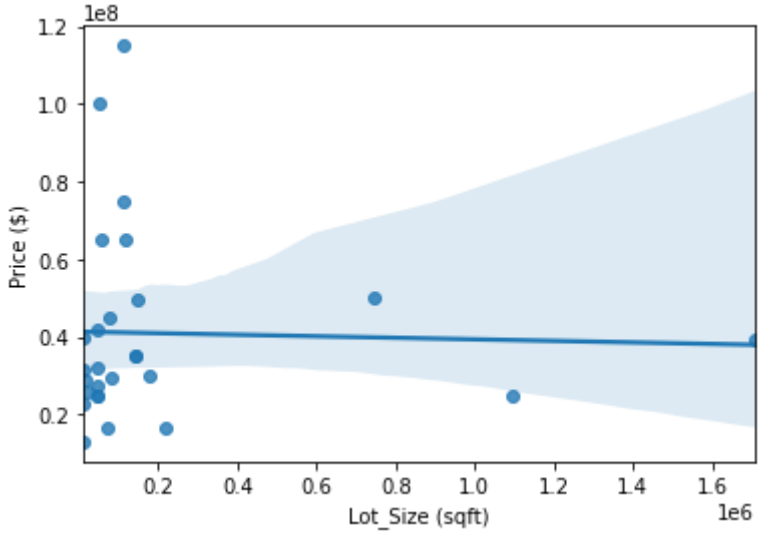
	Location	Price ($)	Bedrooms	Bathrooms	Lot_Size (sqft)	Year Built	min_price($)	max_price($)
0	27930 Pacific Coast Hwy, Malibu, CA 90265	115,000,000.00	12	14	111,784.00	1997	12,300,000.00	21,530,000.00
1	31118 Broad Beach Rd, Malibu, CA 90265	100,000,000.00	5	14	51,382.00	2020	10,650,000.00	21,470,000.00
2	24186 Case Ct, Malibu, CA 90265	75,000,000.00	5	7	113,678.00	2021	21,042,310.85	23,013,076.92
3	22102 Pacific Coast Hwy, Malibu, CA 90265	22,950,000.00	4	4	11,137.00	1000	8,590,000.00	15,700,000.00
4	33256 Pacific Coast Hwy, Malibu, CA 90265	65,000,000.00	3	4	114,569.00	1000	4,100,000.00	6,180,000.00

Visualization of the dataset

In [ ]:
```
sns.regplot(malibu_dataset['Lot_Size (sqft)'], malibu_dataset['Price
 ($)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

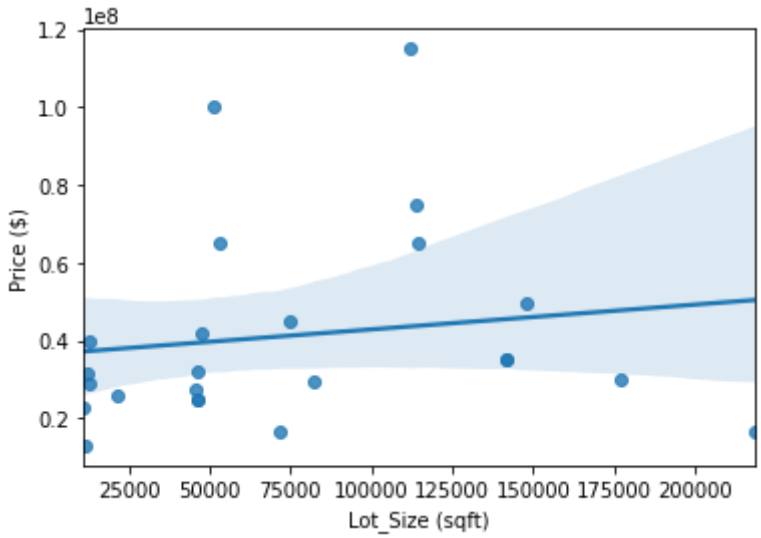
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd9264db190>



In [ ]:
```
z = malibu_dataset[malibu_dataset['Lot_Size (sqft)']<600000]
sns.regplot(z['Lot_Size (sqft)'], z['Price ($)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

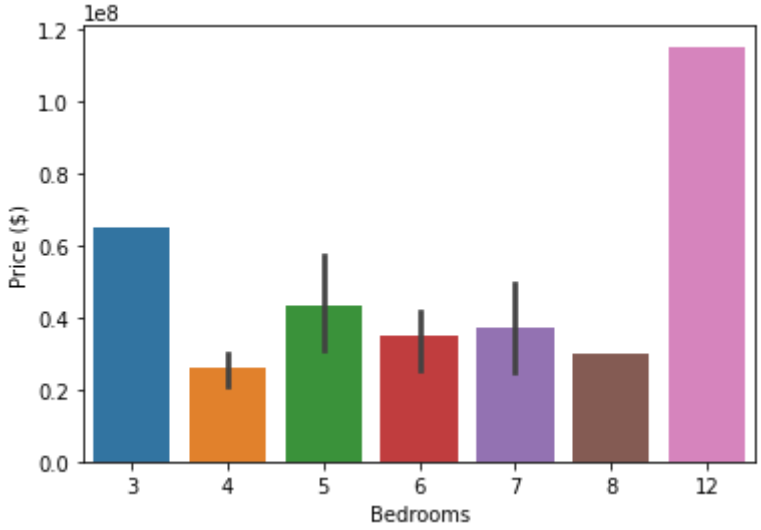
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd925c9bb10>



In [ ]:
```
import seaborn as sns
sns.barplot(malibu_dataset['Bedrooms'], malibu_dataset['Price ($)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

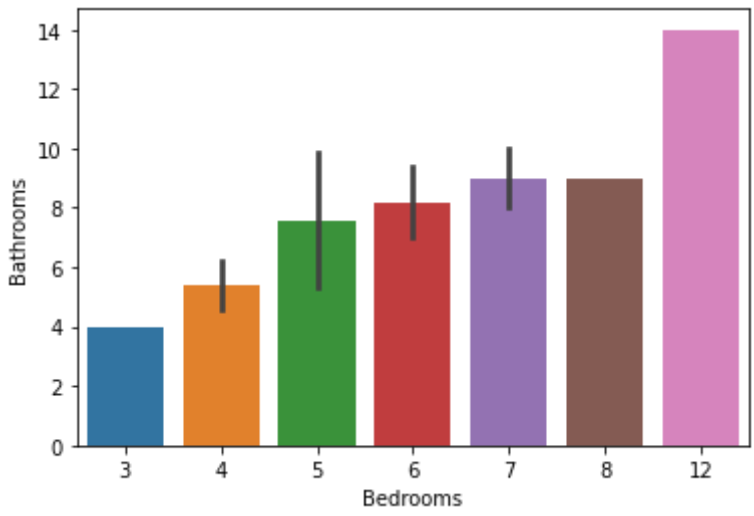
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd925463410>



In [ ]:
```
sns.barplot(malibu_dataset['Bedrooms'], malibu_dataset['Bathrooms'],ci=
'sd')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

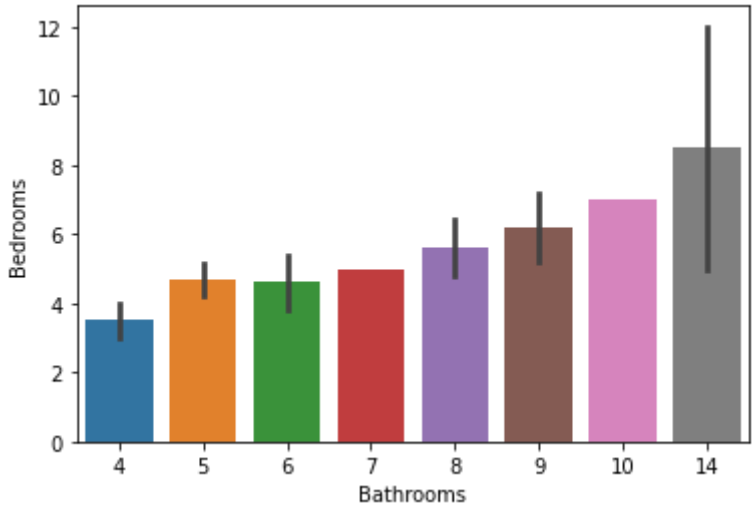
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd9253c0cd0>



In [ ]:
```
sns.barplot(malibu_dataset['Bathrooms'], malibu_dataset['Bedrooms'], ci=
'sd')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

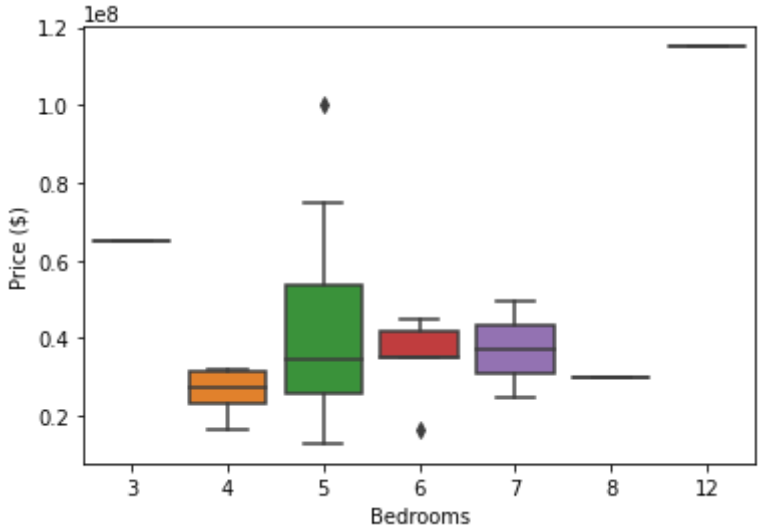
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd9263c9f50>



In [ ]:
```
sns.boxplot(malibu_dataset['Bedrooms'], malibu_dataset['Price ($)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

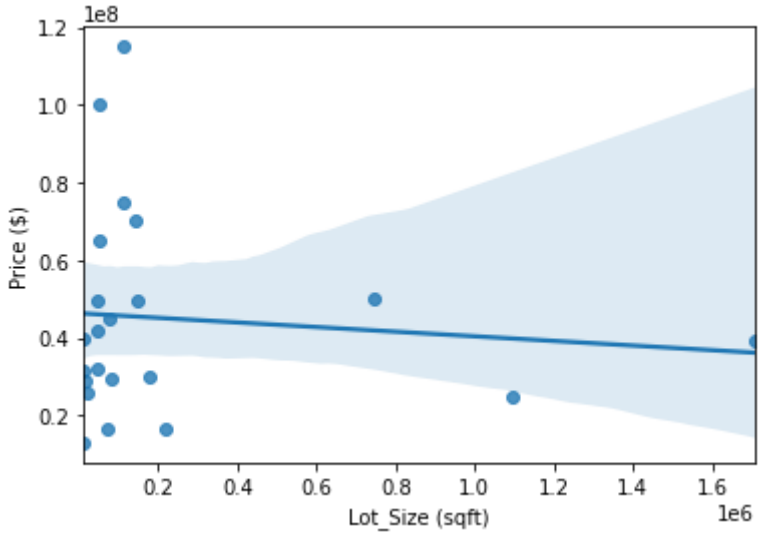
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd9252df350>



In [ ]:
```
z = malibu_dataset[malibu_dataset['Year Built'] > 1000]
z = z.groupby(['Year Built','Lot_Size (sqft)'],as_index=False)['Price
 ($)'].sum()
sns.regplot(z['Lot_Size (sqft)'], z['Price ($)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futur
eWarning: Pass the following variables as keyword args: x, y. From vers
ion 0.12, the only valid positional argument will be `data`, and passin
g other arguments without an explicit keyword will result in an error o
r misinterpretation.
  FutureWarning
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd92515dc90>



One Hot Encoding of 'Year Built'

In [ ]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers = [('encoder', OneHotEncoder(), [5
])], remainder = 'passthrough')
malibu_dataset1 = ct.fit_transform(malibu_dataset)
```

In [ ]:
```
pd.DataFrame(malibu_dataset1).head()
```

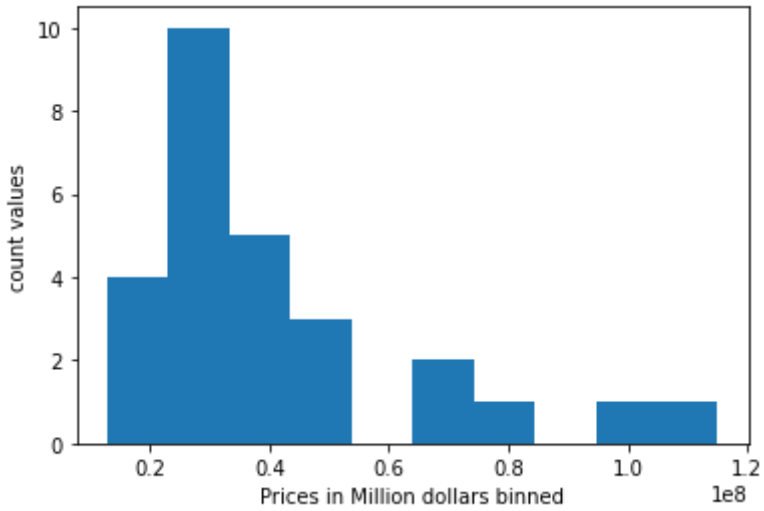
Out[ ]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
3	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0
4	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0

Checking for normal distribution in the target variable 'Price"

- Histogram Plot
- P-test

In [ ]:
```
plt.hist(np.array(malibu_dataset['Price ($)']))
plt.xlabel("Prices in Million dollars binned")
plt.ylabel("count values")
plt.show()
```



In [ ]:
```
from scipy import stats
k2, p = stats.normaltest(malibu_dataset['Price ($)'])
```

In [ ]:
```
p
```

Out[ ]:
```
0.00035155348142355765
```

The value of p is close to 0, the hypothesis is rejected for a normal distribution.

To make the distribution closer to normal, we can either apply log, log(1+x) or box-cox transform

Checking Pearson Correlation between all variables

In [ ]:
```
sns.heatmap(malibu_dataset.corr())
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd925015dd0>
```